

NASA Contractor Report 189557

1N 61
61446
P-35

Technical Requirements Analysis and Control Systems (TRACS)
Initial Operating Capability (IOC) Documentation

Dana P. Hammond

Computer Sciences Corporation
Hampton, Virginia

Prepared For
Langley Research Center
under Contract NAS1-19038
November 1991



National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

(NASA-CR-189557) TECHNICAL REQUIREMENTS
ANALYSIS AND CONTROL SYSTEMS (TRACS) INITIAL
OPERATING CAPABILITY (IOC) DOCUMENTATION
(Computer Sciences Corp.) 35 p CSCL 09B

N92-15604

Unclas
63/61 0061446

Technical Requirements Analysis and Control System (TRACS)
Initial Operating Capability (IOC) Documentation

November 1991

	Page
Table of Contents	i
1. INTRODUCTION	1-1
1.1 Identification	1-1
1.2 Scope	1-1
1.3 Purpose	1-1
1.4 Document Organization	1-1
1.5 User Profile	1-2
2. SYSTEM OVERVIEW	2-1
2.1 Application Objectives	2-1
2.2 Design Objectives	2-2
2.3 Design Approach	2-2
2.4 Component Overview	2-3
2.5 User Interface	2-3
2.6 Data Base Component	2-6
2.7 Report Capability	2-7
2.8 Hardware and Software Requirements	2-7
3. SYSTEM OPERATION	3-1
3.1 Logging In and Logging Out	3-1
3.2 Using the Menu Structure	3-3
3.3 Loading and Unloading Data	3-5
3.4 Entering Data	3-7
3.5 Querying Data	3-7
3.6 Deleting and Updating Requirements	3-8
3.7 Generating Reports	3-9
3.8 Error Recovery	3-9
4. Glossary and Acronyms	4-1
4.1 Definitions	4-1
4.2 Acronyms	4-2
Appendices	
A. Database Schema and Linkage	
B. List of Menus	

List of Figures**Page**

2-1	Overview of TRACS components	2-4
2-2	HyperCard fundamentals	2-5
2-3	Overview of the Oracle DBMS	2-6
2-4	TRACS report generation	2-8
3-1	TRACS screen hierarchy	3-2
3-2	Log on Card	3-3
3-3	Main Menu	3-4
3-4	System Administration Menu	3-5
3-5	Requirements Menu (a)	3-7
3-6	Requirements Menu (b)	3-7

1.0 INTRODUCTION

Section 1 identifies the document, its scope, its purpose, its organization, and other introductory information for all readers.

1.1 Identification

The Technical Requirements Analysis and Control Systems (TRACS) software was developed to supplement the effort of project status and change management. This is the sole document describing all aspects of TRACS, including scope and purpose, system description, and system operation.

1.2 Scope

The Technical Requirements Analysis and Control Systems provide supplemental tools for the analysis, control, and interchange of project requirements. This effort addresses the sharing and control of information on a project-wide basis (i.e., not limited to an individual location). Project requirements defines both allocated and derived requirements which initially consists, but is not limited to, the statement of work.

Although TRACS was designed and developed for Earth Observing System (EOS) instruments (i.e., SAGE III, CERES etc.), the resultant system can be applied to a variety of related projects.

1.3 Purpose

The purpose of the Technical Requirements Analysis and Control Systems (TRACS) is to provide all qualified project users with direct access to pertinent project information, including resource requirements and requirement status. TRACS supports research projects by providing the user community with a focal point for project requirements.

Additionally, TRACS provides the capability to analyze and control requirements. TRACS provides the users with a system that supports efficient and consistent operations, integration, and accommodation of requirements.

TRACS uses relational data base technology in a stand-alone or in a distributed environment that can be used to coordinate the activities required to support a project through its entire life-cycle.

TRACS utilizes a set of keyword and mouse driven screens which imposes adherence through a controlled user interface. The user interface provides the users with an interactive capability to interrogate the data base and to display or print project requirement information. In addition, TRACS will generate specified reports to facilitate analysis.

1.4 Document Organization

This document reflects both the user view and the implementer's view of the requirements.

Section 1 provides a general overview to TRACS and its heritage.

Section 2 provides a system-level (high-level) identification of the basic processes and interfaces within the system. This section describes the application and design objectives, lists alternative

approaches, and provides a detailed description of the underlying components.

Section 3 describes the system operation, detailing the data required, system events and functional interrelationships.

Section 4 is a standard document section which provides supplementary Glossary and Acronyms.

1.5 User Profile

TRACS was designed principally for systems managers, software engineering managers, and lead engineers responsible for the development and maintenance of project requirements.

This document assumes the user is familiar with the basic operations of the Macintosh. For example, the user must know how to invoke a Macintosh application, traverse through cards, and perform mouse operations. For further information concerning Macintosh topics, see the HyperCard User's Guide or the HyperCard Handbook.

2. System Description

This section describes the Technical Requirements Analysis and Control Systems (TRACS) as a system, focusing on the application objectives, the design objectives, and the hardware and software configuration.

2.1 Application Objectives

The Technical Requirements Analysis and Control Systems (TRACS) serves as a common basis for change control, analysis, and the interchange of information. While TRACS is intended to supplement project management efforts, it is not intended as a configuration management system, nor to replace other activities such as scheduling and planning. TRACS' intended scope is reflected in the breadth and depth of information collected. TRACS provides the method of maintaining and analyzing requirements, but does not necessitate how it should be used.

The use of the TRACS serves as a common basis of information interchange, analysis and control. This system ensures:

- the appropriate and adequate set of analysis operations are available,
- the pertinent project information has been identified,
- the format in which the information is collected and stored is consistent, and
- the organization of requirement information supports a common and coherent analysis of data.

The TRACS provide the connectivity and tools for the interchange of project requirements that needs to be shared and controlled on a project-wide basis. Through the use of TRACS, requirements can be accessed from the local site or a copy of the information can be sent to remote users having the TRACS software package implemented on their local system.

The requirement information collected is focused on change-control, requirement status and traceability. A detailed description of the requirement information is provided in the database schema and the user interface described in the remaining sections.

The information collected primarily consists of the following related areas:

Requirement description

- requirement identification
- requirement text
- date requirement entered or modified
- requirement origin
- requirement subsystem (used to identify related areas of concern)
- keywords (used to group requirements for traceability)
- requirement comments

Verification description

- planned verification method
- actual verification method
- verification status
- verification date
- verification comments

To support change-control, a history of both requirements and verification is kept in the form of revision numbers. The initial requirement is entered into the database, along with associated characteristics, and regarded as "revision 1 of 1". Any change in the requirement or verification is collected and represented as the next revision (i.e., 2 of 2). The user interface allows the users the choice to query either the latest revision of each requirement or all revisions.

In addition to developing and maintaining requirements, the above basic information can be used to perform the following analysis:

- track project progress and completion in terms of individual requirement, subsystems, and interrelated requirements,
- identify which requirements can be attributed to a set of engineering efforts,
- attach other pertinent data to a requirement to aid in analysis (i.e., cite a reference for derived requirements, detail the rationale for a requirement change, etc.),
- indicate how requirements have been satisfied and associated information, and
- support assessment of changes to requirements.

2.2 Design Objectives

Several design considerations were incorporated in the development of the Technical Requirements Analysis and Control Systems. These considerations include:

- the identification of the major processes needed to satisfy the application objectives,
- the selection of a design approach to embody the major processes,
- the identification of components to accomplish the objectives,
- the ability of the components to expand with growth or adapt with other configurations, and
- the configurability of the system to meet the needs of specific projects.

2.3 Design Approach

There are four common approaches in handling requirement traceability. This section summarizes these approaches, and concludes with the method used to implement TRACS.

The first approach, Traceability Matrix, consists of a text editor maintaining a list or matrix describing the requirements and interested attributes. This approach has a low start-up cost, but is limited in maintaining changes and analysis capability.

The second approach, consists of maintaining the requirements with a formal DBMS. This method combines a friendly user interface with either a single user or multi-user DBMS. This combination represents a common and inexpensive platform on which to build requirements traceability tools. The DBMS can be run stand-alone on a PC or on a multi-user workstation for handling larger projects. Additionally, on a workstation, this tool can be integrated with other resources. This approach is often tailored for one host (PC or workstation), and difficult in writing a consistent user interface.

The third approach, deriving database from tags inserted in source documents, allows users to start with the original requirements and identify how records are to be allocated. This approach allows tracing back to the original requirement documentation, but does not support change management nor analysis.

The final approach, an integrated system incorporating a set of requirements traceability tools, is typically a combination of the previous approaches. This method has a more expensive initial start-up cost, but provides tools for immediate use. This approach provides a tool which may be difficult to tailor to a specific project, and may include features not used.

The initial application consists of a limited set of requirements, a rapidly changing interface, and a small number of users. The approach selected for TRACS combines a flexible user interface in conjunction with a relational DBMS. Currently, the DBMS is designed for a single user, but can be modified to incorporate multiple users. The current combination was selected for the ease in prototyping a management system for the targeted application. The result of this effort allows for the determination of functionality and capabilities needed in future projects.

The selection of versatile components was a key design consideration, as to not restrict the system to a single set of machines, or range of analysis and output capabilities.

2.4 System Components

TRACS is composed of three major components: a user interface, a relational database management system, and a report generator. Figure 2-1 is an overview of TRACS illustrating the internal components and their interrelationships. The following sections provide a brief description of each components' function.

2.5 User Interface

The TRACS user interface consists of the set of cards and menus presented to the user when TRACS is invoked. This interface provides the conceptual framework in which the user interacts with the system. The user interface performs a variety of functions including: controlling the flow of the program, providing a consistent interface through the operations and data offer, ensures consistencies by limiting choices, and focuses the user on the application by relevant commands.

The TRACS user interface is designed to hide the implementation issues from the user, and focus the user on requirement activities. For example, even though TRACS uses a database management system, the user interface makes no direct references the DBMS. Thus, the process of maintaining and manipulating the data could be replaced without an impact on the use of the interface. Note however, the interface coordinating the efforts of the underlying tools, such as the DBMS, would be impacted.

Macintosh

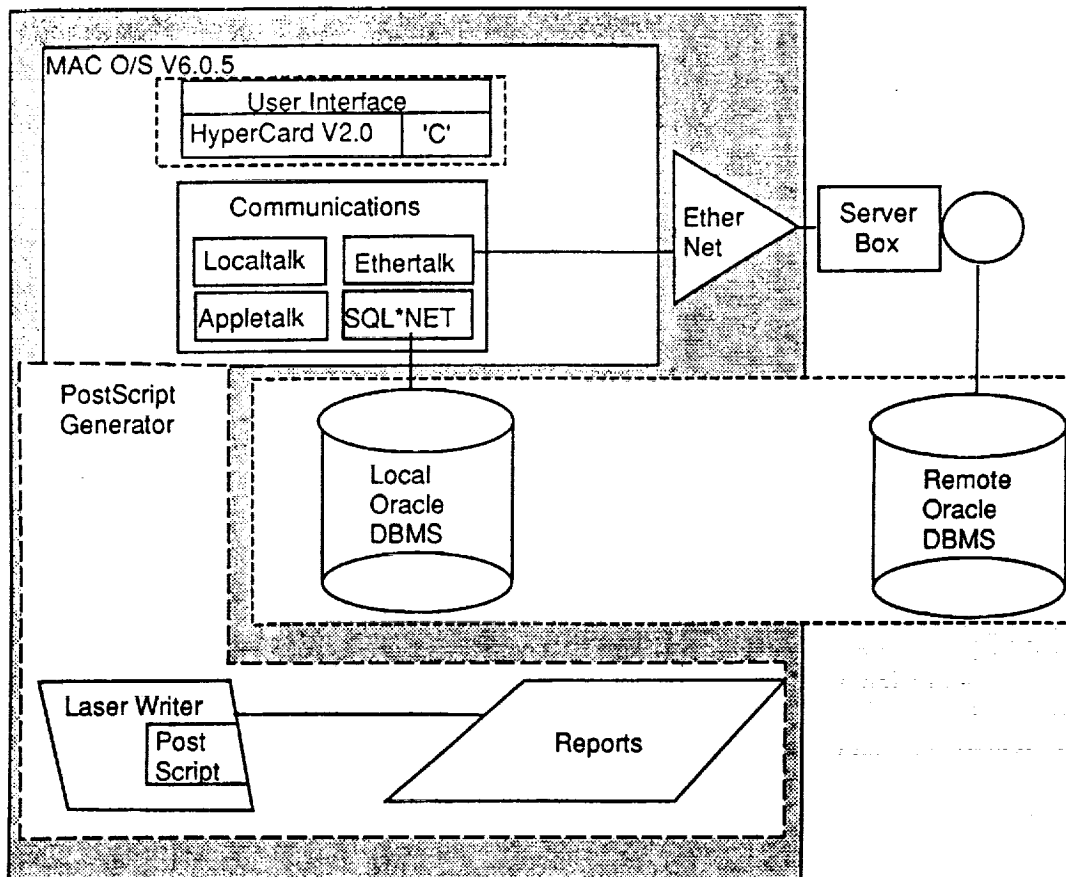
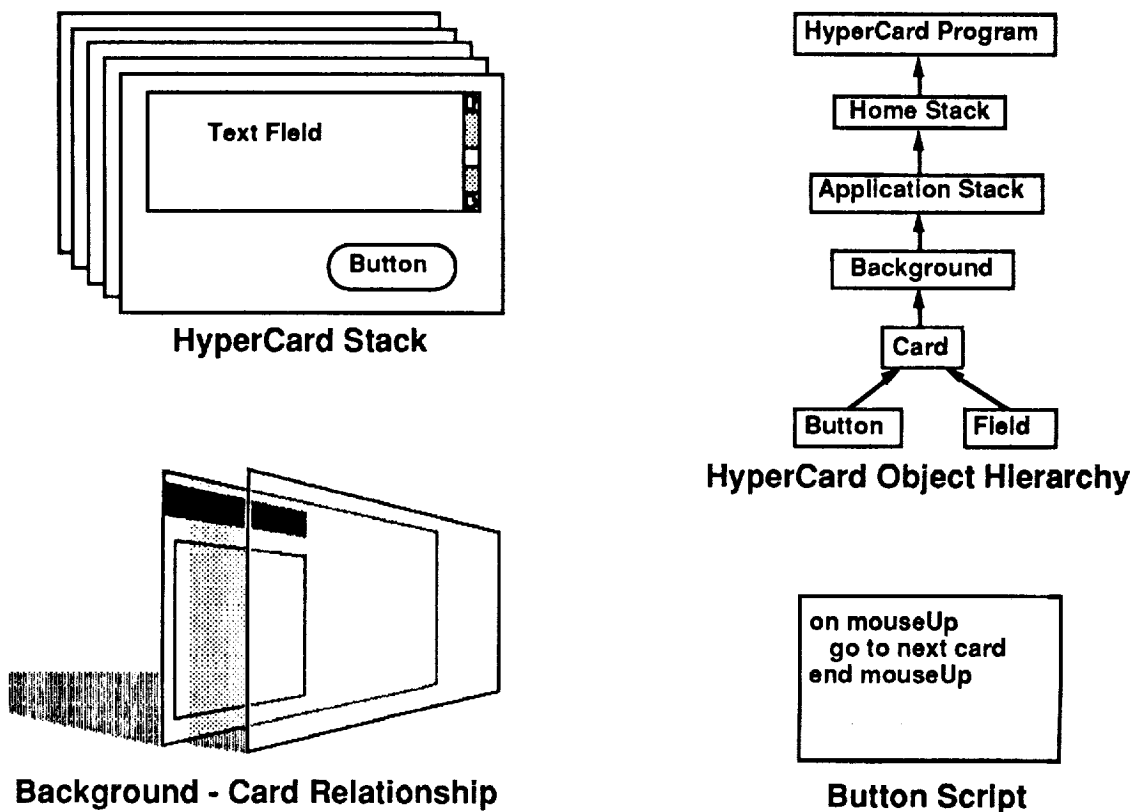


Figure 2-1. Overview of TRACS components.

The TRACS interface is a combination of HyperCard and 'C' programming language. HyperCard is available with all Macintoshes, and provides a quick prototyping capability. HyperCard offers the ability to create a set of cards on which fields and buttons can be organized. Figure 2-2 shows some of the fundamental constructs supported by HyperCard. Each card, button, and field have "message handlers" which can allow for a set of operations to be performed.

Although HyperCard is limited to the Macintosh family, HyperCard emulators and look-alikes are becoming available to support non-Macintosh machines.

HyperCard Fundamentals



CSC 03-19-91 TRACS

Figure 2-2. HyperCard fundamentals.

2.6 Data Base Management System (DBMS)

TRACS uses relational data base technology to manage and query all requirements data. The amount of information and types of queries require the use of a formalized DBMS. The DBMS provides a wide range of functions including: a data definition language (DDL) for describing the data to be stored, a data manipulation language (DML) for efficient data operations (selects, deletes, updates, etc.), and methods of data base support features.

The Oracle DBMS was selected for its ease of use in the Macintosh environment, its availability on other platforms (i.e., MicroVax workstations, Convex, etc.), and its data base management constructs. Figure 2-3 shows an overview of the Oracle DBMS. Oracle can be used to interface with a local database (i.e., on the same machine), or transparently with a remote database across a network.

The DBMS features addressed include:

- data management: separation of physical storage from the way users view the data,
- concurrency issues: dead lock detection and prevention, data integrity between operations,
- tools to access data and develop applications: access via C, SQL, or supported applications,
- efficient transaction environments: buffering of data, fine tuning operations for specific needs,
- security: database access or database objects.

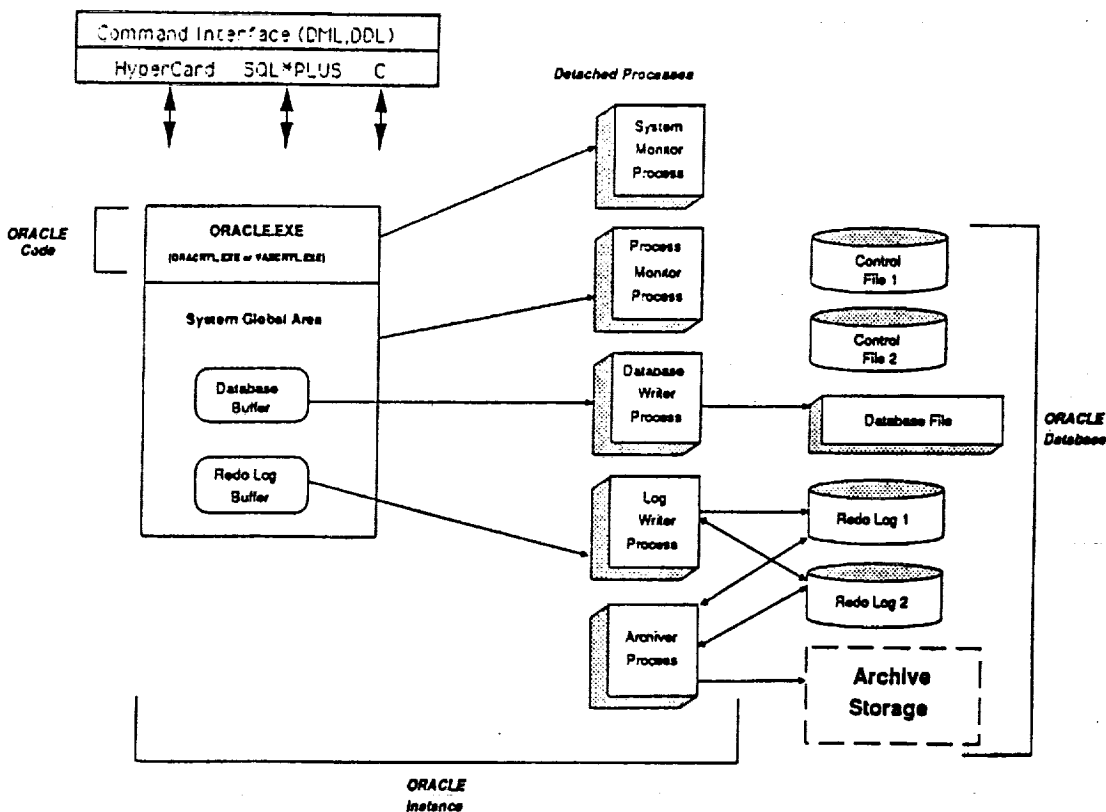


Figure 2-3. Overview of the Oracle DBMS.

2.7 Report Generation

The ability to generate reports is necessary to facilitate requirement documentation and analysis. PostScript, and interpreted language, was chosen for its flexibility in generating output, and its availability on laser writers. PostScript is the name of a computer programming language developed to communicate high-level graphic information to digital laser printers. Figure 2-4 shows the ability of TRACS to generate reports interactively or save reports to a file for future printing. The PostScript format is readily interpreted on a large variety of hardcopy devices.

2.8 Hardware and Software Requirements

This section briefly describes the hardware and software requirements necessary to load and execute the TRACS.

2.8.1 Hardware Requirements

TRACS is currently implemented for the Macintosh series: Macintosh SE, Macintosh SE/30, Macintosh II, Macintosh IIfx, Macintosh IIfx, Macintosh IIfx, Macintosh IIfx, or Macintosh Portable. You may also use a Macintosh Plus capable of running System 6.0 and Finder 6.1

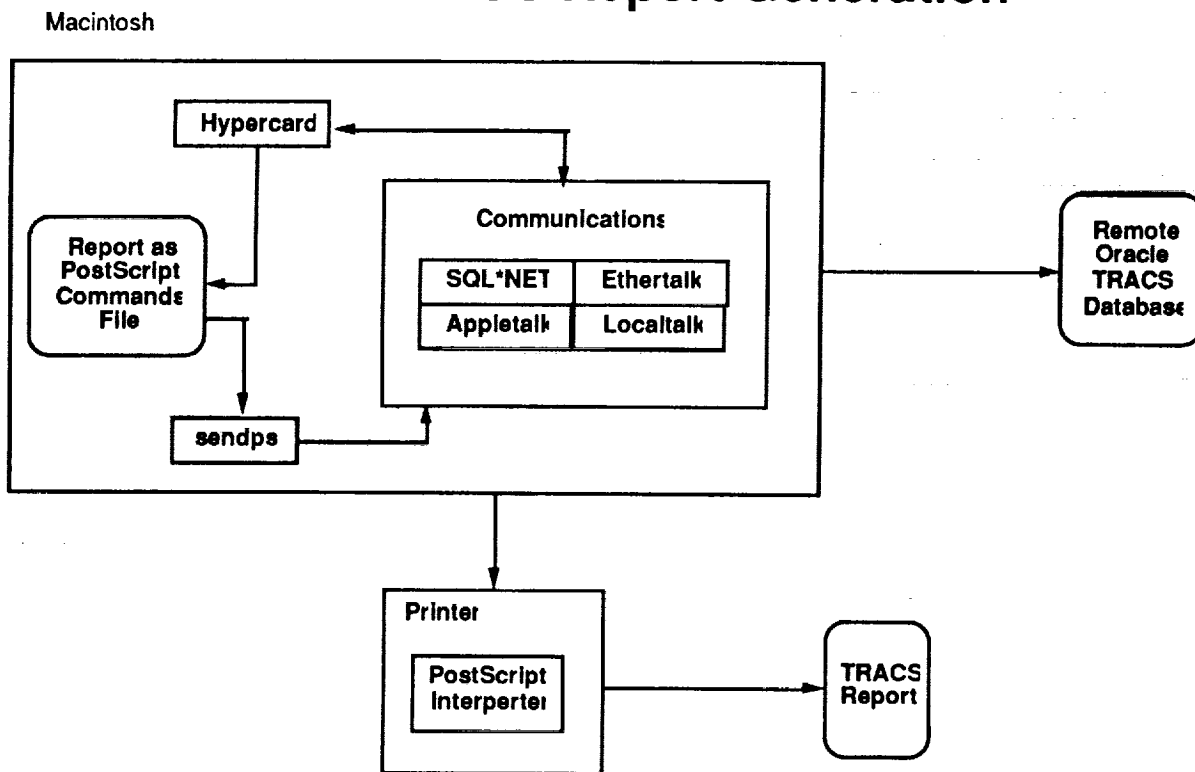
TRACS and underlying database, i.e., ORACLE, requires a hard disk with at least 6 megabytes of available disk space, and at least 2 megabytes of available memory. Although it is preferable to run TRACS under MultiFinder, if your Macintosh has less than 1 megabyte of internal memory (RAM) remaining after starting with MultiFinder, run TRACS under Finder.

2.8.2 Software Requirements

The version numbers of the software you can use with TRACS for Macintosh are listed below:

- System, version 6.0 or later
- Finder or MultiFinder, version 6.1 or later
- HyperCard, version 1.2 or later
- Oracle 1.2 or later

TRACS Report Generation



CSC 03-19-91 TRACS

Figure 2-4. TRACS report generation.

3. System Overview

This section provides an overview of Technical Requirements Analysis and Control Systems' (TRACS) capabilities and operations. This overview will familiarize the user with how to log in and log out of the system, how to traverse through the menus, how to load and unload data, how to enter and query data, and how to generate reports.

TRACS combines a friendly user interface, designed to facilitate requirement management, with a database management system. The user interface is database independent, providing the user with a mechanism to access the requirement information without knowledge of the database. This separation allows the user to focus on the requirement activities while allowing the interface to handle the database activities. Because of this database independence, future updates of the database, or even the use of a different database, will be transparent to the user. However, once data is entered into a specific database, it will require an effort to move a user to another database software system.

The central element of the TRACS is the description of the requirements. Each requirement consists of a set of optional and necessary information. The requirement information is entered through the Requirements Menu, ensuring data consistency, correctness, and completeness.

Since some information are constituent parts of the requirement, for example "responsible person", they must already exist in the database and available through the other support menus for use by the Requirements Menu. To enter subordinate information, select the appropriate "update" button from the Main Menu. Another menu, specific to the type of information being entered, will be provided to the user. Figure 3-1 shows the TRACS screen hierarchy. Once entered, this subordinate information is available for the construction of the requirements in the Requirements(a) menu.

Once a set of requirements have been entered in the underlying database, the user can query the database. The Requirements Menu, allows the user to enter the requirement information, querying this information from the database, and delete and modify entries.

TRACS Screen Hierarchy

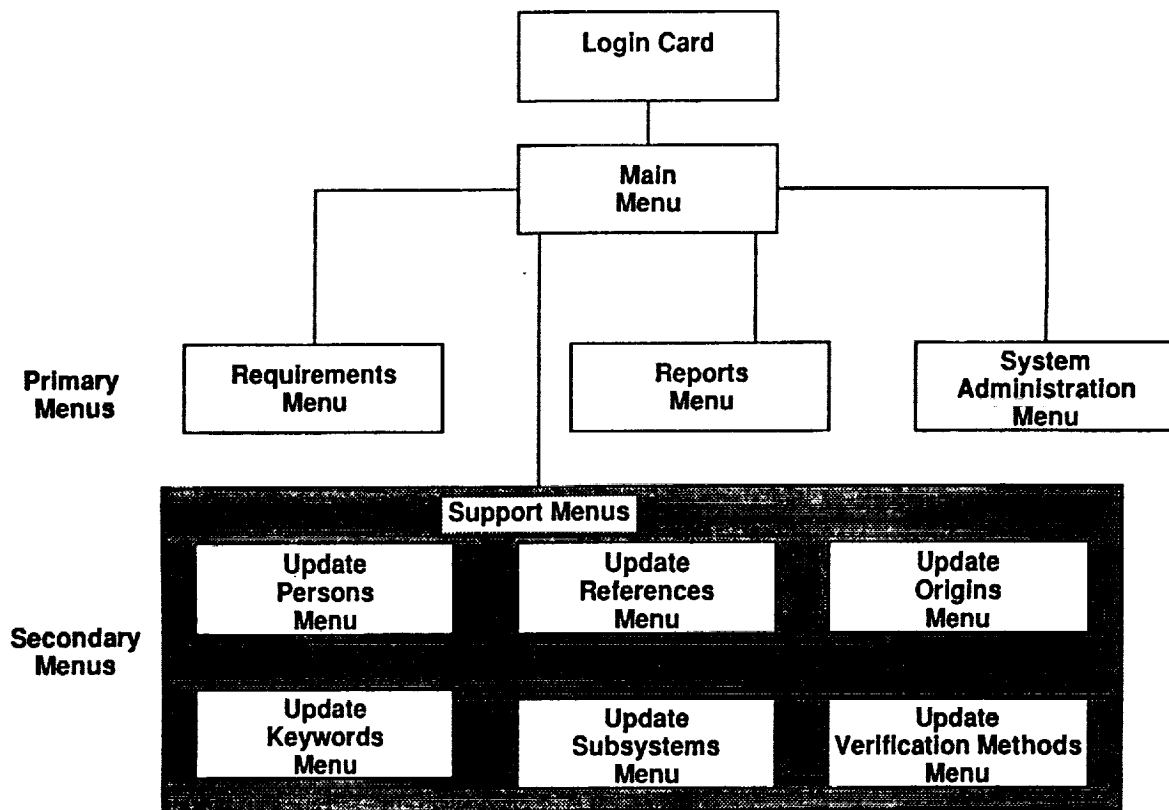


Figure 3-1. TRACS screen hierarchy.

3.1 Logging In and Logging Out

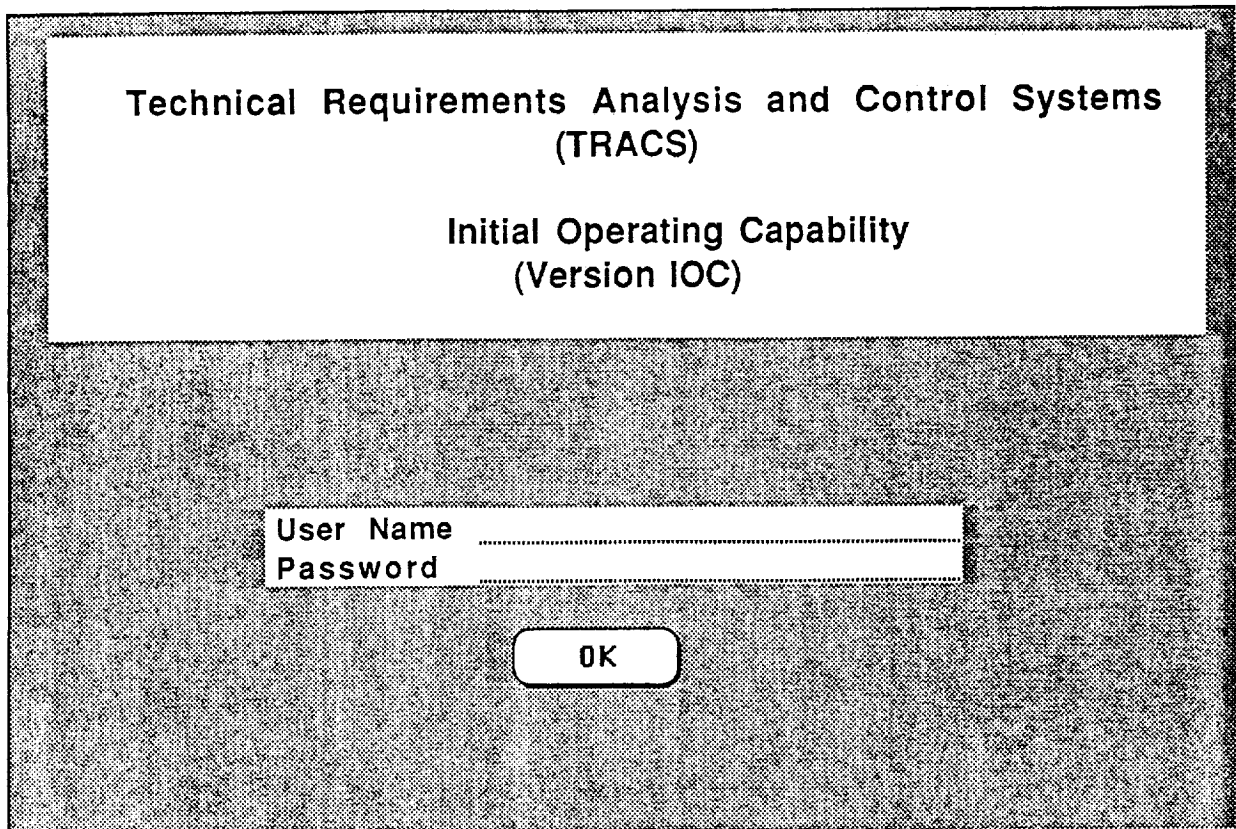
Before starting the TRACS the user must ensure the ORACLE database is properly installed and initialized. When the Macintosh is powered up or restarted, the ORACLE icon should appear during machine initialization. The presence of the icon verifies that ORACLE has been installed properly, and is being initialized. If the icon is not displayed, do not invoke the TRACS, for the TRACS program may be corrupted.

The user invokes the TRACS by selecting the TRACS DB folder and invoking the TRACS icon. This begins execution of the TRACS and displays the LOG ON card (see figure 3-2). This card contains two fields: User Name and Password. You should have received a valid User Name and Password when the program was distributed to you.

The LOG ON card ensures only valid users may access the requirements database. Additionally, each user is associated with a limited set of operations which can be performed using the TRACS.

To access the database, enter your User Name and Password. If this information is valid, then the Main Menu (see figure 3-3) will appear. If your log in information is invalid, then you will be prompted to try again or exit the program.

Once successfully logged into the system, the user can log out by selecting the "Log Out" button in the lower-right corner of the Main Menu. This will return you to the Macintosh desk top environment.



Technical Requirements Analysis and Control Systems
(TRACS)

Initial Operating Capability
(Version IOC)

User Name

Password

OK

Figure 3-2. LOG ON Card

3.2 Using the Menu Structure

Once the valid user information is entered, the underlying database is started and attached to your application, and the Main Menu is displayed. The Main Menu serves as a central point to access all other TRACS activities through a series of "buttons" (see figure 3-3). To invoke a button, merely move the mouse to the button and select. The current screen will be replaced by another menu of operations. Each of these menus will return you to the Main Menu. Appendix B has a complete list of all menus referenced throughout this document.

Main Menu

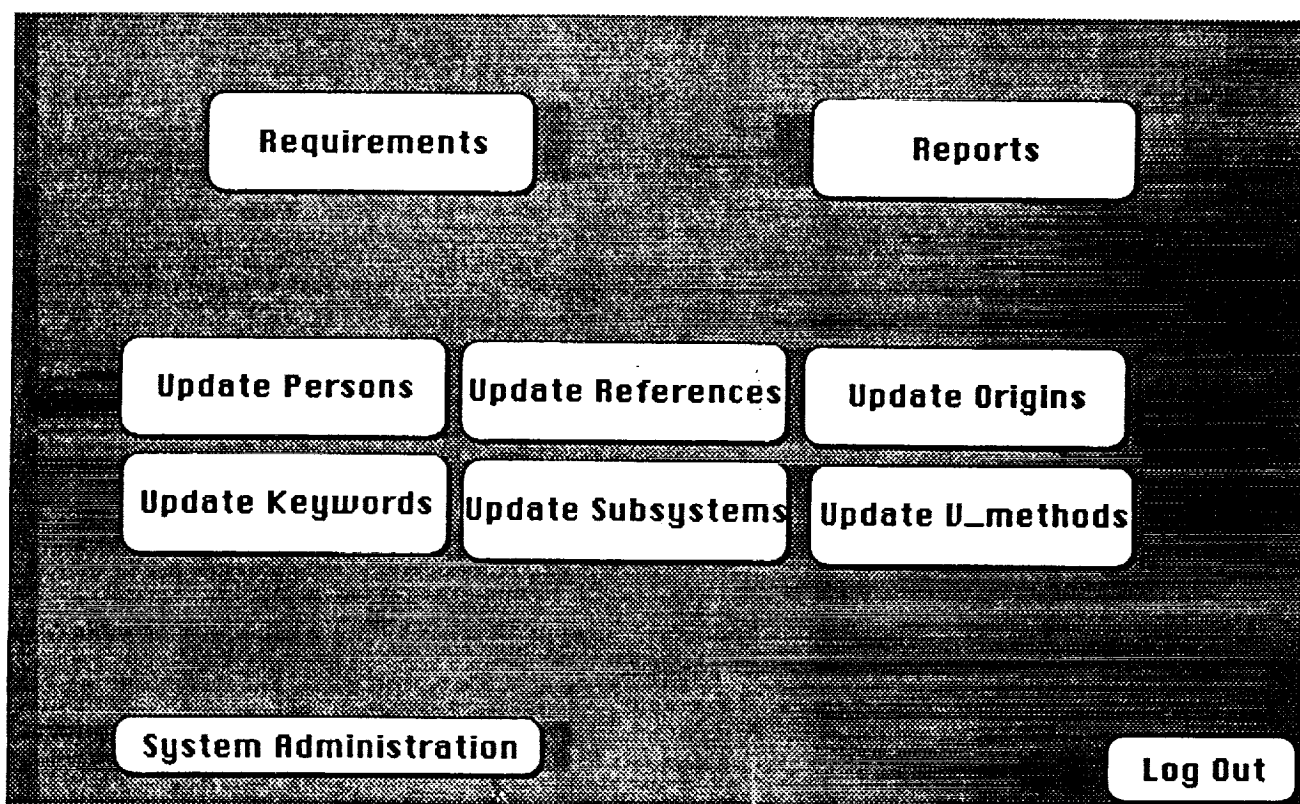


Figure 3-3. Main Menu

3.3 Loading and Unloading Data

The TRACS provides the ability to "unload" the current database from a floppy disk or another folder on the harddisk into a file, and "load" a database from a file to a floppy disk or another folder on the harddisk through the System Administration Menu (see figure 3-4). Unloading allows the user to maintain a backup copy of the database. Also the user can ship a copy of the database to a remote user having TRACS on their Macintosh.

To "unload" a database, enter the path of the file name into the upper box. Be sure to include the drive name and the full path name. Then select the "Unload database" button. All the information and the schema in the database will be written to the named file. A message to the user that the unload is complete will appear on the screen.

To "load" a database, enter the path of the file name into the upper box as before. Then select the "Load database" button. The program will prompt to overwrite the existing database. If you choose to overwrite, the current database's data and schema will be deleted, the schema will be installed, and data from the file will be loaded. If you choose not to overwrite, the information in the file will be written into the current database. If the schema differs between the current database and that on the file, then depending on the discrepancies, data may be lost.

To delete all the information in the current database, select the "Delete rows" button. This will remove all data from the current database, but leave the database schema intact.

To count the number of records in the current database, select the "count" button. This will display the count of all records in the current database. This button is useful in validating the number of entries written into and subsequently read from the database files.

If an error occurs, a message with a negative number will appear in the field in the center of the System Administration Menu screen and/or in the message box. Write down the information and context in which the error occurred and report it to the support group. Ensure the full file name is valid (including blanks and underscores).

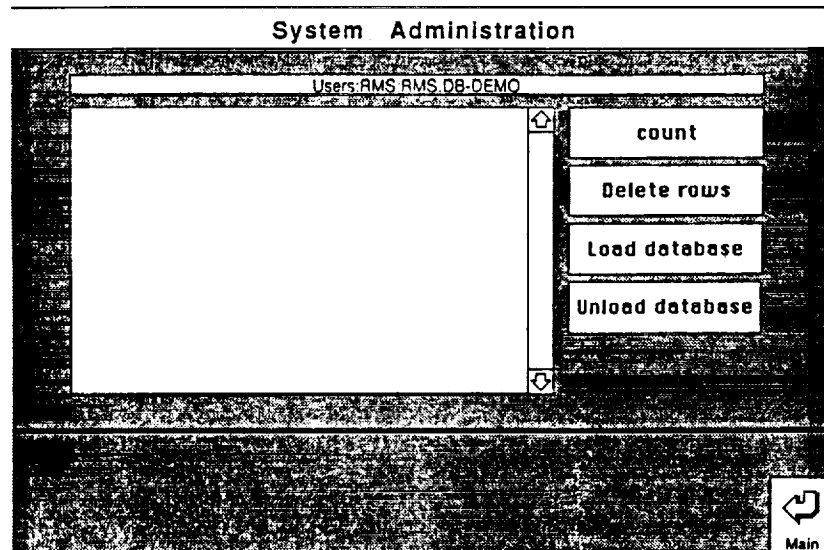


Figure 3-4. System Administration Menu.

3.4 Entering Data

The Requirements Menu (see figures 3-5 and 3-6) consists of two cards: Requirements(a) comprising of a set of fields in which the user enters the data using a select method from lists of data existing in the database, and Requirements(b) consisting of three fields to enter the requirement text information. To enter data in the various fields of card a, move the cursor over the field and click. If the field has a limited set of values, a list with these values will be displayed, and the user will be able to scroll and select a value. For field with no lists, the user must enter text directly.

To enter data into 'card b', the user should select the down arrow button in the lower-right corner of 'card a'. This will display 'card b'. The fields will be set to line one, in which the user can enter the text information. To return to 'card a' from 'card b' select the up arrow button in the lower-right corner.

Once the user has entered all pertinent data, then select the insert button. If any required information is missing, the field(s) will be underlined and a message displayed. The user must enter data in the underlined fields. Once all fields are filled, select the insert button again. The cursor will change to a "busy" cursor until the operation is complete. The data has been entered into the current database.

3.5 Querying Data

To find requirements in the database which match a specified criteria, the user will use the fields of the Requirements Menu. When the Requirements Menu is first displayed, all the fields will be empty. The user can enter the field(s) with values, the same way as entering data, then select the "Find" button. The number of requirements found will be displayed in the bottom-right corner next to the Main Menu return button.

If any requirements match all the fields selected, the number of records will be displayed (called "records found"), and a list of requirements will be retrieved (called "found records list") and made available for viewing. The number of "records found" will be presented, and three arrows (left, right, and bottom) will be displayed surrounding the requirement index number. The index number will initially be set to 0, showing the fields used to make the query.

Selecting the right arrow will increase the index and display the requirement information pointed to by the index. Note, the text information for that requirement is also available in 'card b' by selecting the large down arrow located in the bottom-right of the card. Selection of the right arrow again will increase the index for the next requirement. If the index number exceeds the number of requirements found, then the 0 index will be displayed again.

Selecting the left arrow will decrease the requirement index in a similar manner. The small down arrow will reset the requirement index to 0 (the initial query information).

To remove a requirement from the "found records list", traverse through the found records until the appropriate requirement is found. Then select the "toss out" button, and the requirement in the "found records list" will be removed, and the "records found" will be decreased.

To make another query, select the "clear" button to empty all the fields on the card.

Requirements (a)

Requirement	El General (1)	<input type="checkbox"/> L	Revision	1	of	1
Subsystem	External Interface	<input type="checkbox"/> =	Revision Date	16-MAR-90		
Origin	Pallet Simulator SW Req	Responsible Person		Jackson, Wade		
Section	4.0 External Interface Req					
Keywords	Real-time constraints					
Verification						
Planned Method	Inspection	<input type="checkbox"/> =		Date Verified	16-MAR-90	
Status	Not tested	Verification Person		Peterson, Doug M.		
Actual Method	Inspection					
Requirement text	Requirement comments		Verification comments			
<div> <div>find</div> <div>insert</div> <div>delete</div> <div>update</div> <div>find tags</div> <div>clear</div> <div>toss out</div> </div>						
delete all		update all		<div> <div>1 Records found</div> <div>↔ 1 ↔</div> <div>↓</div> </div>		
			<div> <div>↓</div> <div>↶</div> <div>Main</div> </div>			

Figure 3-5. Requirements(a) Menu

Requirements (b)

Requirement text:	
The Pallet Simulator will duplicate all of the Smart Flexible Multiplexer Demultiplexer/TITE Instrument Interfaces (SDIO, PBD, DILs, and DOLs).	
Requirement comments:	
The related timing diagrams are included in the Appendices B, C, and D of the Pallet Simulator Specifications.	
Verification comments:	
<div> <div>↑</div> <div>↶</div> <div>Main</div> </div>	

Figure 3-6. Requirements(b) Menu

By default the "find" button only selects the "latest" revision of the requirements. To find "all" requirements, or requirements based on a date, select the "find options" button. A set of tags (boxes) will be displayed on the screen with default values. The tag by the "revision" field will have a "L", the default value, meaning only the latest revisions will be found. If an "A" is picked into the tag, then all revisions conforming to the selection criteria will be returned.

The tags associated with the date fields allow the user to find requirements based on a date. If no date is entered, then the date fields are ignored. If a date is present, then the operation in the tag will be applied (i.e., before the date, before and on the date, etc.).

3.6 Deleting and Updating Requirements

To delete or update an existing requirement in the database, the user must first select the requirement. That is, the user must use the "find" button in the Requirements Menu as described above. Then, use the small arrows to select the requirement index to operate on. If a requirement is to be deleted, then selecting the "delete" button will attempt to delete the requirement on the screen. If the requirement is either the "latest" (i.e., the last revision) or the only revision, then a message will be displayed. Once selected, the requirement will be deleted from the database and from the "found records list".

To update an existing requirement in the database, again "find" the requirement and move to the correct index. Now the user can change the information in the current requirement's fields as before. Once all the data for 'card a' and 'card b' have been updated, the user selects "update". If all required fields are satisfied, then the requirement will be updated in the database. The new requirement will not be part of the "found records list", but will reside in the database. The current requirements in the "found records list" will have their revision numbers changed to reflect the updated number of revisions.

3.7 Generating Reports

Reports are performed in a similar fashion as the requirements(a) menu.

3.8 Error Recovery

The user interface examines all the data entered to ensure requirement consistency and correctness. If an inappropriate value is entered, an error message will be displayed. The user is forced to acknowledge these messages by selecting the presented options (sometimes only one).

If an internal error occurs and is detected by the system, an error message will be displayed, and the prompt "Report" will be issued. Please write down the current information and context in which the error occurred, and relay the information on to the support group.

4.0 GLOSSARY

4.1 Definitions

attribute - An inherent characteristic generally used in describing an external appearance of a requirement.

baseline - A stable point in a project where the effort accomplished is accepted as correct and serves as a basis for further development.

current database - The set of requirements and related records in the user's local database.

defaults - A set of initial values for a set of attributes.

found records list - A subset of requirements selected from the database through the "find" button on the "requirements(a)" menu.

latest requirement - The last or latest revision of a specific requirement. For example, revision 2 of 2.

requirement - A qualitative description of an aspect of what the system will accomplish. A requirement is verifiable and has an associated set of attributes.

specification - A quantitative description of an aspect of what the system will accomplish.

4.2 Acronyms

<u>DB</u>	Data Base
<u>DBMS</u>	Data Base Management System
<u>EOS</u>	Earth Observing System
<u>IOC</u>	Initial Operating Capability
<u>LaRC</u>	Langley Research Center
<u>NASA</u>	National Aeronautics and Space Administration
<u>RAM</u>	Random Access Memory
<u>SAGE III</u>	Stratospheric Aerosol and Gas Experiment III
<u>TRACS</u>	Technical Requirements Analysis and Control Systems

APPENDIX A

Database Schema and Linkage

Appendix A contains a diagram representing the TRACS Database Linkage, and several pages consisting of the the database schema used to describe the underlying database.

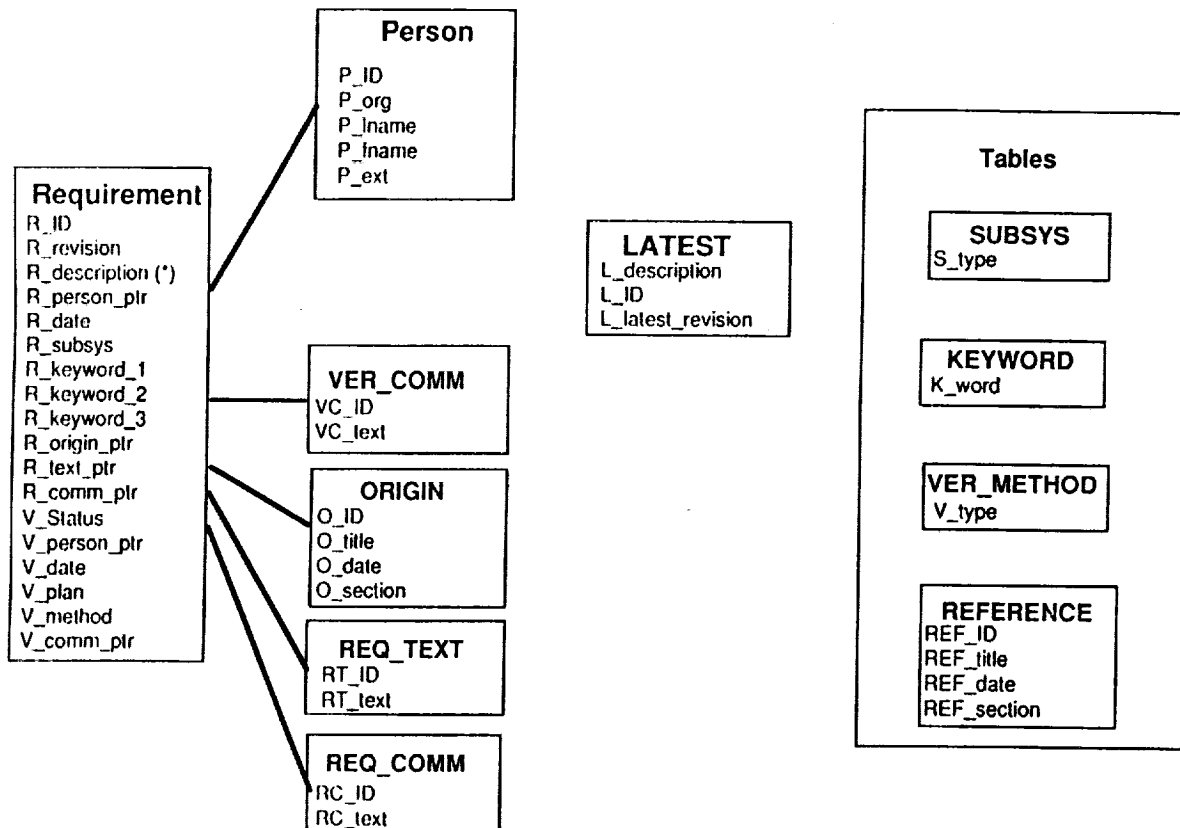
The Database Linkage consists of a set of tables represented by boxes containing the table fields. Tables are linked together with solid lines depicting a pointer (or index) from one table to another. The LATEST table is an internal table used and maintained by the TRACS. The tables grouped together in the Tables box, form a collection of supplementary information which supports the Requirement table, but are not "pointed" to directly. Note, these table names are consistent with those in the following database schema.

The database schema is a description to the underlying format database, detailing all aspects of the schema. The tables are presented here to the user in order to obtain field limitations (i.e., limited to 40 characters).

TRACS Database Linkage
TRACS Database Schema

PAGE
A-2
A-3

TRACS Database Linkage



CSC 03/19/91 TRACS

```
CREATE TABLE REQUIREMENT(
```

```
!
```

```
! The REQUIREMENTS table is the basis for the SAGEIII RMS providing the
! foundation for all other information.
```

```
!
```

R_ID	NUMBER	NOT NULL,	!* KEY: Unique requirement ID (Internal)
R_REVISION	NUMBER	NOT NULL,	!* KEY: Unique revision number for R_ID
R_DESCRIPTION	CHAR(40)	NOT NULL,	!* KEY: Description to facilitate unique recognition
R_PERSON_PTR	NUMBER	NOT NULL,	!** POINTER: Person information (PERSON.P_ID)
R_DATE	DATE	NOT NULL,	! Entered or modified date
R_SUBSYS	CHAR(40)	NOT NULL,	! Subsystem type (from SUBSYS table)
R_KEYWORD_1	CHAR(40),		! Keyword 1 (from KEYWORD table)
R_KEYWORD_2	CHAR(40),		! Keyword 2 (from KEYWORD table)
R_KEYWORD_3	CHAR(40),		! Keyword 3 (from KEYWORD table)
R_ORIGIN_PTR	NUMBER	NOT NULL,	!** POINTER: Origin document (ORIGIN.ORIGIN_ID)
R_TEXT_PTR	NUMBER	NOT NULL,	!** POINTER: Requirement text (REQ_TEXT.RT_ID)
R_COMM_PTR	NUMBER	NOT NULL,	!** POINTER: Req. Comment (REQ_COMM.RC_ID)
V_STATUS	CHAR(1)	NOT NULL,	! Verification status: Passed, Failed, Not tested
V_PERSON_PTR	NUMBER	NOT NULL,	!** POINTER: Verification person (PERSON.P_ID)
V_DATE	DATE	NOT NULL,	! Verification date
V_PLAN	CHAR(40)	NOT NULL,	! Planned verification method (VER_METHOD table)
V_METHOD	CHAR(40)	NOT NULL,	! Actual verification method (VER_METHOD table)
V_COMM_PTR	NUMBER	NOT NULL);	!** POINTER: Ver. comments (VER_COMM.VC_ID)

```
!
```

```
.....
```

```
!
```

```
CREATE TABLE PERSON(
```

```
!
```

```
! The PERSON table is a static table of person referenced by SAGEIII RMS.
```

```
!
```

P_ID	NUMBER	NOT NULL,	!* KEY: Unique PERSON ID (Internally used)
P_ORG	CHAR(20)	NOT NULL,	! Organization name
P_LNAME	CHAR(20)	NOT NULL,	! Last name
P_FNAME	CHAR(20)	NOT NULL,	! First name
P_EXT	CHAR(20)	NOT NULL);	! Phone number

```
!
```

```
.....
```

```
!
```

```
CREATE TABLE REQ_TEXT(
```

```
!
```

```
! The REQ_TEXT table contains requirement text entries.
```

```
!
```

RT_ID	NUMBER	NOT NULL,	!* KEY: Unique REQ_TEXT identifier (Internal)
RT_TEXT	LONG);		! Requirement text

```
!
```

```
.....
```

```
!
```

```

CREATE TABLE REQ_COMM(
!
! The REQ_COMM table contains requirement comment entries.
!
RC_ID          NUMBER    NOT NULL,    ! * KEY: Unique REQ_COMM identifier (Internal)
RC_TEXT        LONG);          ! Requirement comments
!
.....
!
CREATE TABLE VER_COMM(
!
! The VER_COMM table contains verification comment entries.
!
VC_ID          NUMBER    NOT NULL,    ! * KEY: Unique VER_COMM identifier (Internal)
VC_TEXT        LONG);          ! Verification comments
!
.....
!
CREATE TABLE VER_METHOD(
!
! The VER_METHOD table contains verification method entries.
!
V_TYPE        CHAR(40)    NOT NULL);  ! Verification method (Unique identifier)
!
.....
!
CREATE TABLE SUBSYS(
!
! The SUBSYS table subsystem type entries.
!
S_TYPE        CHAR(40)    NOT NULL);  ! Subsystem type (Unique identifier)
!
.....
!
CREATE TABLE KEYWORD(
!
! The KEYWORD table contains keyword list entries.
!
K_WORD        CHAR(40)    NOT NULL);  ! Keyword entry (Unique identifier)
!
.....
!
CREATE TABLE ORIGIN(
!
! The ORIGIN table contains statement of work entries.
!
O_ID          NUMBER    NOT NULL,    ! * KEY: Unique Origin identifier
O_TITLE       CHAR(40)    NOT NULL,    ! Origin title
O_DATE        DATE       NOT NULL,    ! Origin version date
O_SECTION     CHAR(40));  ! Origin section description
!
.....

```

```

CREATE TABLE LATEST(
!
! The LATEST table contains a list of the latest REQUIREMENTS indices.
!
L_DESCRIPTION CHAR(40) NOT NULL,    ! * KEY: Description to facilitate unique recognition
L_ID           NUMBER  NOT NULL,    ! Unique requirement ID (Internal)
L_LATEST_REVISION NUMBER NOT NULL); ! Latest revision number for R_ID
!
!.....
!
CREATE TABLE REFERENCE(
!
! The REFERENCE table contains reference list entries.
!
REF_ID          NUMBER  NOT NULL,    ! * KEY: Unique Reference identifier
REF_TITLE       CHAR(40) NOT NULL,    ! Reference title
REF_DATE        DATE    NOT NULL,    ! Reference version date
REF_SECTION     CHAR(40));           ! Reference section description
!
!.....

```


APPENDIX B

Description of Menus and their Function

This appendix contains a complete list of the TRACS menus, and a brief description of each functionality:

<u>Menu/Card</u>	<u>Description</u>
Log On Card	verifies valid user information, and loads the database.
Main Menu	serves as a centralized menu to access all support menus.
Requirements	accesses the Requirements Menu allowing the user to enter, query, modify, and delete requirement attributes and related text in the current database.
Reports	accesses the Reports Menu allowing the user to query data and generate a report.
Update Persons	accesses the Update Persons Menu allowing the user to enter, query, modify, and delete persons information.
Update References	accesses the Update References Menu allowing the user to enter, query, modify, and delete reference information.
Update Origins	accesses the Update Origins Menu allowing the user to enter, query, modify, and delete origin information.
Update Keywords	accesses the Update Keywords Menu allowing the user to enter, query, modify, and delete keyword information.
Update Subsystems	accesses the Update Subsystems Menu allowing the user to enter, query, modify, and delete subsystem information.
Update V_methods	accesses the Update Verification Methods Menu allowing the user to enter, query, modify, and delete verification method information.
System Admin	provides a mechanism to 'load' and 'unload' the database.
Log Out Card	provides the means to log out of TRACS.

Technical Requirements Analysis and Control Systems (TRACS)

Initial Operating Capability
(Version IOC)

User Name
Password

OK

Main Menu

Requirements

Reports

Update Persons

Update References

Update Origins

Update Keywords

Update Subsystems

Update U_methods

System Administration

Log Out

Requirements (a)

Requirement El General (1) ☐ Revision 1 of 1
 Subsystem External Interface ☐ Revision Date 16-MAR-90
 Origin Pallet Simulator SW Req Responsible Person Jackson, Wade
 Section 4.0 External Interface Req

Keywords Real-time constraints

Verification

Planned Method Inspection
 Status Not tested ☐ Date Verified 16-MAR-90
 Actual Method Inspection Verification Person Peterson, Doug M.

Requirement text	Requirement comments	Verification comments
------------------	----------------------	-----------------------

find	insert	delete	update
find tags	clear	toss out	

<input type="button" value="delete all"/>	<input type="button" value="update all"/>	1 Records found ↩ 1 → ↓	↓ ↺	↻ Main
---	---	-------------------------------	--------	-----------

Requirements (b)

Requirement text:	↑
The Pallet Simulator will duplicate all of the Smart Flexible Multiplexer Demultiplexer/TITE Instrument Interfaces (SDIO, PBD, DILs, and DOLs).	↓
Requirement comments:	↑
The related timing diagrams are included in the Appendices B, C, and D of the Pallet Simulator Specifications.	↓
Verification comments:	↑
	↓
	↑ ↻ Main

Update Persons

Name (last, first MI.)	Organization	Phone
Franks, Joyce	Structural Analysis	(813) 722-0982
Jackson, Wade	Mechanical Engineeri	(804) 865-1725
Jackson, Wake R.	Science and Technolo	(804) 827-0202
Peterson, Doug M.	Applied Technology	(804) 864-1991
Spain, Charles	Instrument Research	(802) 965-1195

Name (last, first MI.)	Eckhause, Julia		
Organization	Aerodynamics Inc.		
Phone	(804) 889-9075	format: (yyy) xxx-xxxx	

Update References

Reference Identifier	Section	Date
Pallet Simulator SW Req Spec	2.1 Product Specification (Volume 1)	01-FEB-90
Product Specification and DIDs	Release 4.3	28-FEB-89
Smart Flexible Multiplexer (SFMDM)	Release 1.0	09-JAN-90
Software Management Plan	1.0 TITE Electronics System	01-SEP-89
TITE Instrument Spec	3.2 SFMDM	16-JAN-90

Reference ID	Pallet Quality Assurance		
Section	Release 2.0		
Version Date	25 May 89	(dd mmm yy)	

Update Origins

Origin Identifier	Section	Date
Pallet Simulator SW Design	4.1 Master AT	03-SEP-90
Pallet Simulator SW Req	4.0 External Interface Req	13-MAY-90
Pallet Simulator SW Req	4.1 Interface Timing Req	13-MAY-90
Pallet Simulator SW Req	4.1.1 Inherent to the SFMDM	13-MAY-90
Pallet Simulator SW Req	4.1.2 Dictated by the TITE Instrument	13-MAY-90
Pallet Simulator SW Req	4.1.3 Interrupt Generator	13-MAY-90
Pallet Simulator SW Req	4.2 Serial Digital Input Output (SDIO)	13-MAY-90
Pallet Simulator SW Req	4.3 Power Control Box (PCB)	13-MAY-90
Pallet Simulator SW Req	4.3.1 PCB Hardware	13-MAY-90
Pallet Simulator SW Req	4.3.2 PCB Software Interface	13-MAY-90
Pallet Simulator SW Req	4.4 Discrete Input Lows (DILs)	13-MAY-90
Pallet Simulator SW Req	5.0 Process and Data	13-MAY-90



Origin Title Pallet Simulator SW Design
 Section 4.2 Actuator AT
 Version Date 03 Sep 90 (dd mmm yy)

Update keywords


Keyword
Array Processor
High-risk
Real-time constraints
Shuttle dependent

Keyword Optical Analysis



Update Subsystems

Subsystem	
Electronics	 
External Interface	
Mechanical	
Off-line Processing	
Optical	
Pointing	
Real-time Processing	
Thermal	


Subsystem Spectrometer

 Main

Update Verification Methods

Verification Methods	
Analysis	 
Combination	
Inspection	
Test	
Vendor data	

Verification Method _____

 Main

System Administration

Harddisk:TRACS_TITE_db_2

Exporting Data...

```

progst = tracs/ball 1024 "Harddisk:TRACS_TITE_db_2" T
YYY REQUIREMENT PERSON VER_COMM ORIGIN REQ_TEXT
REQ_COMM LATEST SUBSYS KEYWORD VER_METHOD
REFERENCE
                    
```


Database unloaded successfully.

count

Delete rows

Load database

Unload database



Main

Reports


Requirement	Revision of
Subsystem <u>Optical</u>	Revision Date
Origin	Responsible Person
Section	
Keywords	
Verification	
Planned Method	Date Verified
Status <u>Not tested</u>	Verification Person
Actual Method	

Reqs by Person	Reqs by Verification Status
Reqs by Keyword	Reqs by Verification Method
Reqs by Date	Reqs by Verification Information
Reqs by Date & Revision Number	Reqs by Ver. Method vs Actual

Sort Attributes

find

clear



Main

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE November 1991		3. REPORT TYPE AND DATES COVERED Contractor Report
4. TITLE AND SUBTITLE Technical Requirements Analysis and Control Systems (TRACS) Initial Operating Capability (IOC) Documentation			5. FUNDING NUMBERS C NAS1-19038 WU 426-43-01-20	
6. AUTHOR(S) Dana P. Hammond				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Computer Sciences Corporation 3217 N. Armistead Avenue Hampton, Virginia 23666-1379			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CR-189557	
11. SUPPLEMENTARY NOTES Langley Technical Monitor: Dr. James R. Schiess (COTR) Lemuel E. Mauldin III				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category 61			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This document describes the Technical Requirements Analysis and Control Systems (TRACS) software package. TRACS offers supplemental tools for the analysis, control, and interchange of project requirements. This package provides the fundamental capability to analyze and control requirements, serves a focal point for project requirements, and integrates a system that supports efficient and consistent operations. TRACS uses relational data base technology (ORACLE) in a stand-alone or in a distributed environment that can be used to coordinate the activities required to support a project through its entire life cycle. TRACS utilizes a set of keyword and mouse driven screens (HyperCard) which imposes adherence through a controlled user interface. The user interface provides an interactive capability to interrogate the data base and to display or print project requirement information. TRACS has a limited report capability, but can be extended with PostScript conventions.				
14. SUBJECT TERMS Computer Programs, Management Systems			15. NUMBER OF PAGES 36	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT	

